A multifactorial evolutionary algorithm for minimum energy cost data aggregation tree in wireless sensor networks

Tran Cong Dao*, Tran Huy Hung*, Nguyen Thi Tam*†, Huynh Thi Thanh Binh*
*Hanoi University of Science and Technology, Hanoi, Vietnam;
†VNU University of Science, Vietnam National University, Hanoi
daotc.bk@gmail.com, tranhuyhung98@gmail.com, tamnt@vnu.edu.vn, binhht@soict.hust.edu.vn

***Abstract*—In wireless sensor networks, the majority of data transmitted by sensor nodes is repeated over and over, and performing processes on them in many cases leads to increased power consumption and reduced network lifetime. Data aggregation is one of the techniques in reducing redundancy and improving energy efficiency; it also increases the lifespan of wireless sensor networks. In this paper, we address the issues of constructing the data aggregation tree that minimizes the total energy cost of data transmissions for two types of networks: without relay nodes and using relay nodes. Traditionally, evolutionary algorithms focus on constructing data aggregation trees for either without relay node networks or using relay nodes networks. Therefore, we show a new approach utilizing a multifactorial evolutionary algorithm, called Potential individuals based Multi-factorial Evolutionary Algorithm (P-MFEA), to solve both issues simultaneously. The proposed scheme shows improved performance in terms of energy consumption.**

***Index Terms*—data aggregation tree, network lifetime, energy efficiency, multifactorial evolutionary algorithm**

I. Introduction

Wireless Sensor Networks (WSNs) are networks of sensor nodes - devices responsible for collecting data from the surrounding environment and sending reports to a base station (also called sink node) for processing. Data from a sensor might be directly sent to the base station or routed through other nodes before reaching the sink. In the Internet of Things era, various applications implementing WSNs have been developed in different areas, such as military applications [1], agriculture measurement networks [2], remote healthcare monitoring [3], and smart home systems [4], etc.

Optimizing energy usage is a crucial challenge for WSNs. Studies in data aggregation have proven to be quite effective to deal with this issue. These algorithms involve collecting data from different sensors into one single node before sending aggregated packets to the base station or another node, which potentially reduces the size of data and minimizes power consumption in the process.

According to [5], approximation algorithms for data aggregation can be mostly divided into four main approaches: centralized, in-network, cluster-based and tree-based. The first two models are simple: In the centralized approach, one header node receive data from all other sensors and performs aggregation, so it has to endure a heavy traffic load. To deal with this issue, the in-network model deploys multiple intermediate nodes to act as aggregators.

In contrast, the latter approaches are complex but flexible. In particular, the cluster-based approach divides the region of interest into multiple clusters. In each cluster, one node is chosen to be cluster head which aggregates data from other nodes of the same cluster. Thus, this model is often used in energy-efficient applications. In [6] and [7], intra and inter cluster aggregation is utilized to compress data and reduce transmission cost. Another approach is used in [8], where cluster size is changed adaptively based on loss ratio metric to increase reliability and reduce consumption. Moreover, Yuea et al [9] introduced a method which divides the network into grids of unequal sizes and schedules cluster heads for consecutive sessions to balance energy usage. In [10], an appropriate method was proposed to aggregate data where firefly algorithm and scheduling based on nodes' activation were used to maintain coverage in WSNs.

Likewise, tree-based is also a popular power-saving approach to WSNs. In this approach, a Data Aggregation Tree (DAT) is first constructed as a base to route data. It could be depicted as a minimum spanning tree with the base station acting as its root. Each node sends data to its parent; hence, data gradually flows from sensor nodes to the base station. Authors of [11] suggested forming a new DAT each round based on the residual energy in order to balance the traffic load between all nodes. On the other hand, in [12], Support Vector Machine is used in DAT to minimize the redundancy and eliminate the false data. Another research from the same authors in [13] focuses on scheduling the process of aggregation by controlling the tradeoff between delay and energy.

Authors of [14] have also applied DAT in two energy optimization models of WSNs: Minimum Energy-Cost Aggregation Tree (MECAT) and Minimum Energy-Cost Aggregation Tree with Relay Nodes (MECAT-RN). Both use a practical aggregation mechanism: The routing of nodes is structured as a DAT. Data in one node is compressed and split into multiple packets that can be transferred periodically to its parent node, thus the connections are stable and a considerable amount of energy is saved. The only difference is that the second model is supported by relay node placement - another well-known technique to support connectivity in WSNs. Authors

proved both problems are NP-complete, which means it is extremely difficult to find the optimal DAT in large instances. They also suggest approximation solutions for both. However, the approximation ratios are still high (2 and 7 for each model, respectively), so the results are rather poor in some scenarios, such as dense graphs or small aggregation ratios.

Tam et al (in [15]) also observed MECAT and offered a Multi-factorial Evolutionary Algorithm (MFEA)-based algorithm called Edge Set Multifactorial Evolutionary Algorithm (ESMFA). MFEA [16] is a state-of-the-art Genetic Algorithm (GA) that focuses on evolutionary multitasking. By representing different task-specific or problem-specific solutions as one individual, GA can be performed on multiple instances of the same problem or multiple distinctive problems concurrently in a unified search space. In recent years, it has emerged as a practical solution for large-scale NP-complete problems. Nonetheless, the effectiveness of MFEA could be further advanced in various aspects: extending for MECAT-RN, exploiting the correlation between two models, and applying existing methods such as 2-approximation (SPT). As a consequence, our study incorporates mentioned ideas to form an improved algorithm which will be detailed in the next section.

The main contributions of this paper are summarized as follows:

- We study the issues of constructing the data aggregation tree that minimizes the total energy cost of data transmissions for two types of networks: without relay nodes (MECAT) and using relay nodes (MECAT-RN). In the WSNs, relay nodes' presence is an extension that improves network connectivity and viability. Because of the coexistence and the similarity between MECAT and MECAT-RN problems in wireless sensor networks, solving these two problems simultaneously is meaningful.
- We propose a new approach utilizing a multifactorial evolutionary algorithm, called P-MFEA, to solve MECAT and MECAT-RN, simultaneously.
- We conduct several simulations to evaluate the performance of the proposed algorithm.

The rest of the paper is organized as follows: Section II presents a basic description of a network model and problem statement. The proposed algorithm is elucidated in Section III. Simulation results and discussions of them are presented in Section IV. Finally, the conclusions and future directions are given in Section V.

II. Problem Statement

A. The Minimum Energy Cost Aggregation Tree problem

In this section, we describe the first problem introduced in [14] - the MECAT. For our use case, we consider a wireless sensor network with n sensor nodes. The communication between two nodes can only be established if the Euclidean distance between them does not exceed the sensor's communication range. Each sensor has some reports to be sent to the base station r (directly or via other nodes). Given a fixed aggregation ratio, Minimum Energy-Cost Aggregation Tree (MECAT) seeks a Data Aggregation Tree (DAT) with minimum energy consumption to route data from all sensor nodes to the base station.

For easier understanding, the model is described as an undirected graph $G = (V, E)$ where:

- $V = \{r, v_1, v_2, ..., v_n\}$ is the set of nodes in which r is the base station and other $v_i \in V(v_i \neq r)$ are n sensor nodes. Each node has a fixed position $v_i = (x_i, y_i)$ in the plane.
- E is the set of edges. Two nodes u and v can transfer data to each other if and only if they satisfy the transmission range constraint $d_{ij} \leq R$, where d_{ij} is the Euclidean distance. In this case, there exists an edge $(u, v) \in E$.
- $s(v_i) \in Z^+$ is the size of reports that sensor node v_i sends to r.
- $q \in Z^+$ is the data aggregation ratio. When u wants to send reports to v, its data is first aggregated and encapsulated into packets. Each packet can contain no more than q units of data.
- T_x and R_x are the energy costs to send and receive one packet, respectively.

MECAT aims to construct a routing tree $T = (V, E_T)$ with $E_T \subseteq E$, so that all sensor nodes can transfer reports to the base r. Provided a routing tree T, each node u receives aggregated packets from descendants and sends them to the parent v. This process continues until all reports are transferred to the root node r.

When transferring packets along an edge $e \in E_T$, both sending and receiving consume energy. Thus, the energy cost for each edge $e \in E_T$ is calculated as follows:

$$C(e) = (T_X + R_X)\lceil \frac{z(e)}{q} \rceil \tag{1}$$

where $z(e)$ is the size of the reports sent along e, and $\frac{z(e)}{q}$ is the number of packets to send after performing data aggregation. Note that each node v_i has to transfer both its own reports of size $s(v_i)$ and data received from its children.

The objective of MECAT is to minimize the total consumption of the routing tree:

$$COST(T) = \sum_{e \in E_T} C(e) \leftarrow min \tag{2}$$

Figure 1 illustrates a simple example of MECAT with 8 sensor nodes.

B. The extended model with relay nodes

The MECAT-RN is directly extended from MECAT. The network model $G' = (V', E')$ is similar to MECAT, except the collection of nodes V' has three types instead of two: A single base station r, a set of sensors U, and some optional relay nodes (which do not have reports on their own and are only responsible for transmitting data). The problem seeks a minimum energy-cost routing tree $T' = (V_{T'}, E_{T'})$, where $V_{T'} \subseteq V'$ and $\{r\} \cup U \subseteq V_{T'}$. Other constraints and the objective remain the same.

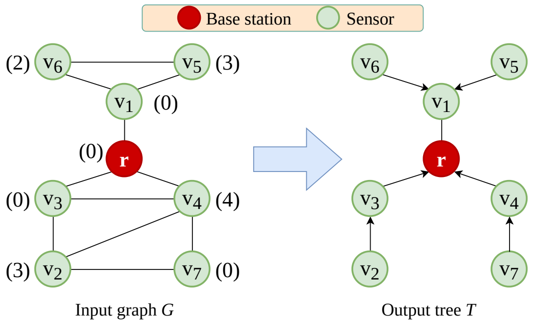


Fig. 1. An example of input and output for MECAT. The size of data for each node is put inside parentheses.

An example of MECAT-RN is shown in Figure 2. The network model G' has 4 sensor nodes and 3 relay nodes. However, the result tree T' only needs to deploy 2 out of 3 relay nodes.

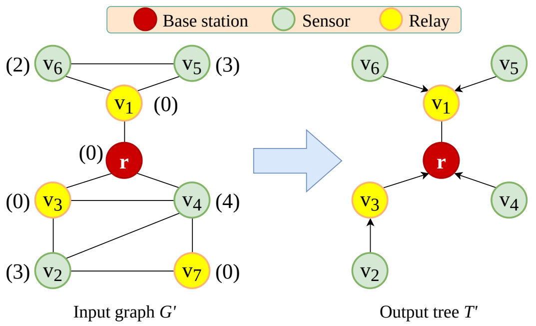


Fig. 2. An example of input and output for MECAT-RN. The size of data for each node is put inside parentheses.

III. Proposed Algorithm

In recent years, MFEA has been proven effective in solving NP-complete problems. In this section, we propose a new approach utilizing MFEA, namely P-MFEA, to solve the MECAT and MECAT-RN simultaneously. Details of the proposed algorithm are presented in the following sections.

A. General scheme of P-MFEA

In MFEA, common knowledge between similar related tasks is used to improve the process of finding solutions. Specifically, transferring information between tasks performed vertically and horizontally can help tasks quickly converge and find the global (or near-global) optimal solutions if the transfer is positive. However, if the search space is not large enough, even getting stuck after some loops can cause the algorithm to converge slowly and obtain a locally optimal solution. To prevent this from happening, we add a number of potential individuals to the search. After the number of loops as Δt, these individuals are added to the current population.

As in the wild, the presence of potential individuals helps population gradually become better. In the context of the evolutionary algorithm, an potential individual can be a promising solution to a problem. We can implement several methods, including local search, approximation algorithm, and greedy algorithm, to create the individuals. The addition of potential individuals to the existing population can enrich the horizontal transfer of knowledge. Moreover, performing genetic operations such as hybridization and mutation with potential individuals will make it possible for P-MFEA to quickly extract useful information from these individuals.

In this study, we use the 2-approximation algorithm in [14] to create potential individuals. Details of the proposed MFEA algorithm are described in Algorithm 1. Besides, our contribution in the framework is highlight in red font.

Algorithm 1: Framework of P-MFEA

Input: K optimization tasks $\{1, \dots K\}$.
Output: The best solutions of all tasks $\{X_k^*, 1 \le k \le K\}$.

```
begin
    Intitialize population P_0 with K × N individuals;
    t ← 0;
    Evaluate P_0 for all the tasks;
    Compute the skill-factor τ_i for each individual in P_0;
    while stopping criterion is not satisfied do
        Offspring population O(t) ← ∅;
        Potential population E(t) ← ∅;
        while |O(t)| < |P(t)| do
            p_a, p_b ← Randomize two parents from P(t);
            Generate a random number rand between 0 and 1;
            if (τ_a = τ_b) or (rand < rmp) then
                o_a, o_b ← Crossover(p_a, p_b);
            else
                o_a ← Mutation(p_a);
                o_b ← Mutation(p_b);
            O(t) ← O(t) ∪ {o_a, o_b};
        if (t > Δt) and (t mod Δt == 0) then
            E(t) ← Creating K × NE potential individuals;
        Update scalar fitness and skill factor for each individual in P(t) ∪ O(t)∪E(t);
        P(t + 1) ← Select the fittest individuals from P(t) ∪O(t)∪E(t);
        t ← t + 1;
```

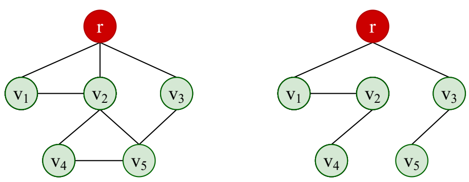
B. Representation of solution in particular space

In this section, we propose a new encoding suitable for each MECAT and MECAT-RN problem, respectively. The solutions to these two problems are in the form of a tree structure. Furthermore, there are several popular tree encryption methods such as Cayley codes, network random key (or netkeys), and edge representation [17]. However, we choose to represent the

solution of the two problems using netkeys encryption because it offers the following benefits:

- The encoding and decoding methods are performed conveniently by well known algorithms in graph to construct a spanning tree such as Prims and Kruskal.
- The individuals obtained by decoding method are always valid. Therefore, there is no need to add individual modification when performing decoding.

At first, the MECAT problem's solution is a spanning tree T of the input graph $G(V, E)$, which will be represented as a vector W with a number of dimensions $|E|$. Each element in W has a real number in the segment 0 to 1 which is mapped to an edge in E, indicating the edge priority when selected to the edge set of the spanning tree T. Figure 3 illustrates an example of encoding strategy for the MECAT problem. In 3(a), an input graph of the problem is given. 3(b) depicts a spanning tree that is a feasibility solution. A chromosome corresponding to the solution is shown in 3(c).



(a) Input graph (b) A feasibility solution

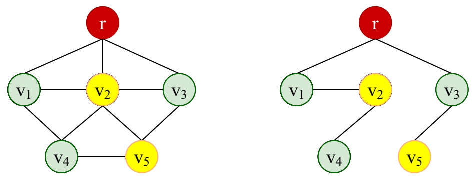
r-v_1	r-v_2	r-v_3	v_1-v_2	v_2-v_4	v_2-v_5	v_3-v_5	v_4-v_5
0.79	0.14	0.64	0.42	0.58	0.21	0.86	0.34

(c) A corresponding chromosome

Fig. 3. Encoding for MECAT problem

However, with relay nodes' appearance in the MECAT-RN problem, encoding this problem's solution is a relatively difficult problem. Unlike the MECAT problem, the problem's solution is a spanning tree, whose vertices may or may not consist of several transition nodes, so the number of vertices of the spanning tree is not fixed. Fortunately, the encoding for the MECAT-RN problem becomes more apparent when we consider a relay node as a sensor node with its own report size of 0. With this in mind, we propose a new encoding based on encryption of random net keys that is consistent with the MECAT-RN problem. In the proposed encoding, the relay nodes appear as sensor nodes. Thus, the encoding of the solution for the MECAT-RN problem will be the same as the MECAT if we consider the problem's solution to be a spanning tree with a vertex set including all sensor nodes and relay nodes. An example of this encoding's illustration is shown in Figure 4.

After presenting the encoding method, a strategy of decoding an individual into a solution to the MECAT and MECAT-RN problem is also proposed. According to the proposed encoding method, the individual decoding for the MECAT and MECAT-RN problem is similar when treating all relay



(a) Input graph (b) A feasibility solution

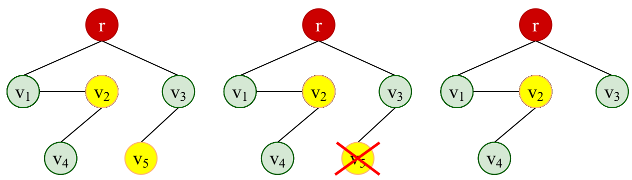
r-v_1	r-v_2	r-v_3	v_1-v_2	v_1-v_4	v_2-v_3	v_2-v_4	v_2-v_5	v_3-v_5	v_4-v_5
0.84	0.11	0.68	0.48	0.34	0.39	0.75	0.43	0.91	0.27

(c) A corresponding chromosome

Fig. 4. Encoding for MECAT-RN problem

nodes as sensor nodes. However, for the MECAT-RN problem, after decoding a solution to a spanning tree, a relay node will be removed from the tree if it is a leaf node. The MECAT-RN problem's solution will be a spanning tree obtained after removing the relay nodes that act as leaf nodes. The reason is that the relay nodes act only to receive and send data between sensor nodes, so these relay nodes cannot be leaf nodes.

Figure 5 describes an example of removing relay nodes in an obtained tree after decoding for the MECAT-RN problem. As can be seen, a spanning tree achieved by decoding an individual in the form of netkeys is shown in 5(a). In this tree, a relay node v_5 is a leaf node; thus, it is deleted from the tree (depicted in 5(b)). Then, a possible tree as a solution for MECAT-RN is illustrated in 5(c).



(a) An obtained tree after decoding the netkeys (b) Remove a relay node was a leaf node (c) A possible tree for MECAT-RN

Fig. 5. Decoding for MECAT problem

C. Individual representation in Unified Search Space (USS)

In the context of MFEA, combining the solution's representation to each problem into a common one in the USS is an essential step. An individual in the joint space must carry the characteristic information of each representation in its own space. Therefore, in our proposal, the representation of individuals in the USS is designed as follows:

- The size of a chromosome in common space is the largest size of chromosomes in individual space. $D = max(D_1, D_2)$ where D_1 and D_2 are the dimensions of solution's representations in the MECAT and MECAT-RN problem, respectively.

- The individual representations with a smaller size than D are padded with zeros to the right.

Then, an individual in USS is represented by a vector W_{uss} of dimensions D. Similar to individual representation in particular space for MECAT, each element in W_{uss} is associated with an edge of the graph in common space. An example of the encoding strategy in USS is depicted in Figure 6.

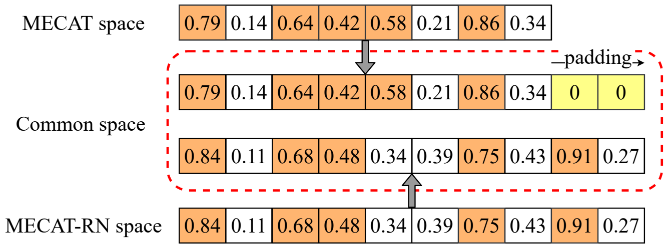


Fig. 6. Representation of two problems in common space

Furthermore, we propose decoding individuals from common space to particular space. For the representation of netkeys presented, the decoding method works as follows: Suppose we have K problems that are solved simultaneously, the dimension of the i^{th} problem is D_i. The individual representation in common space is a real number vector with $D = max(D_1, D_2, ..., D_K)$. The individual representation of the i^{th} problem is defined as the first D_i element in the common representation. Figure 7 illustrates an example of a decoding procedure from USS to separate space.

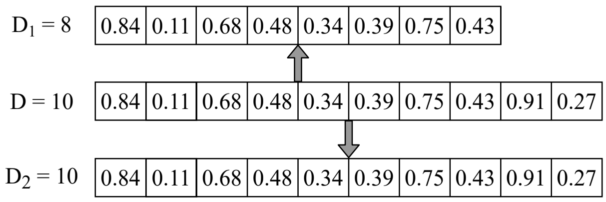


Fig. 7. An example of decoding from USS space to seperate spaces

D. Individual initialization

According to the proposed individual representation, every individual in USS is in the form of vector W_{uss} with D dimensions. Hence, the initialization of an individual is performed by creating an array of D elements. Each element in the array is a real number generated randomly, which follows uniform distribution $U(0, 1)$.

E. Genetic operators

In this section, we describe the following three genetic operators: selection, hybridization, and mutation. In these operators, the selection operator is independent of all representations; vice versa, both the hybrid and mutation operators are specific to each representation type.

In selection, we choose tournament selection to select parents during the selective breeding process. Besides, the elitism selection strategy is applied to select the fittest individuals retained in the next generation population.

Algorithm 2: Proposed crossover operator

Input: Two parent individuals p_1 and p_2.
Output: Two children individuals o_1 and o_2.

begin
  $i, j \leftarrow$ random two points cut $(0 < i < j < D)$;
  $p_1^{'}, p_2^{'} \leftarrow$ the gene segment between positions i and j in p_1, p_2, respectively;
  $o_1, o_2 \leftarrow SBX\_Crossover(p_1, p_2)$ on $p_1{'}, p_2{'}$;
  return two individuals o_1, o_2;

According to the genotype's characteristics in our proposal, the chromosome is represented as a real number vector, so we propose a new hybridization based on SBX [18] and TPX [19]. For each representation in the form of netkeys, instead of applying the SBX crossover operator on all the chromosome elements, we only apply it on the part chosen from the two-cut hybrid. This combination can reduce the likelihood that children resemble a parent entirely caused by high redundancy in netkeys representation. Details of the proposed hybridization operator are described in Algorithm 2.

Moreover, the polynomial mutation is also applied to create new individuals during reproduction.

IV. Computational Results

In this section, we conduct experiments to assess the effectiveness of the proposed algorithm for solving two problems of MECAT and MECAT-RN in terms of averaged objective value, best found objective value, convergence trend, and running time.

A. Experimental datasets

In our experiments, we created two datasets to evaluate the effectiveness of the proposed algorithm. The first dataset for the MECAT problem was generated according to the study [15]. In a wireless sensor network, sensor nodes are randomly placed in a target area. According to the size of the deployment area, the test instances are classified into two types, **Type 1** (area of medium size) and **Type 2** (area of large size). All parameters of these datasets are listed in detail in Table I. In the second dataset for the MECAT-RN problem, each node in the network has a probability of being a sensor node and a relay node of 0.3 and 0.7, respectively. The other parameters of the dataset are the same as the ones for the MECAT problem.

For the convenience of the experiments, we formatted the datasets' names as m_n and l_n corresponding to medium and large instances, where n is the number of nodes in the wireless sensor network.

B. Experimental settings

To verify the performance of our proposal, four algorithms, including 2-approximation (SPT), 7-approximation (LAST) in [14], and multifactorial evolutionary algorithm

TABLE I
DATASET PARAMETERS

Parameter	Value
number of nodes	$100, 110, 120, ..., 190$
size of deployment area ($m \times m$)	$100 \times 100, 200 \times 200$
transmission range (R)	20
aggregation ratio (q)	4
transmission energy constant (T_x)	0.3
receiver energy constant (R_x)	1
non-uniform report size	1, 2, 3, 4, 5
probability of being relay nodes	0.3

(ESMFA) in [15] were used as baseline algorithms. For the MECAT problem, we compare P-MFEA to algorithms SPT and ESMFA; while it is compared to LAST for the MECAT-RN problem. The efficiency of multifactorial evolutionary algorithm for solving MECAT problem when compared to a single-task evolutionary algorithm has been shown in the study [15]. Therefore, we only compare our proposed algorithm to a version of MFEA called ESMFA.

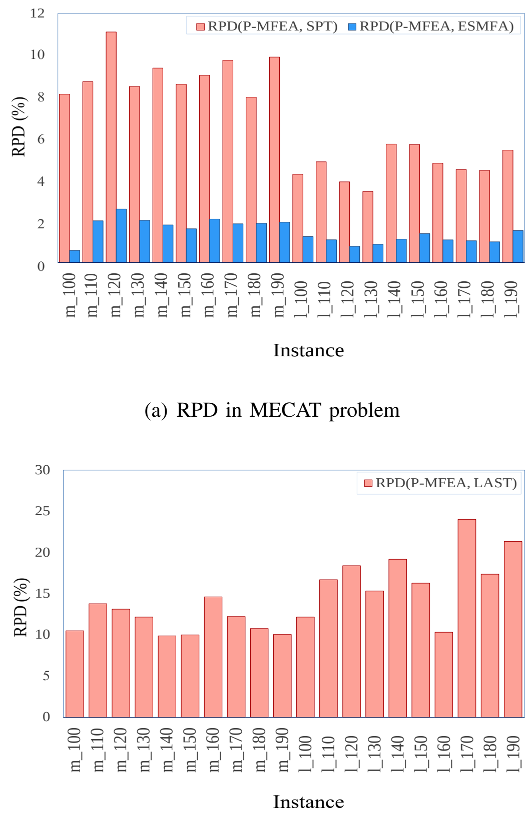
We focus on the following criteria to assess the proposed algorithm's performance: average objective value (***Avg***), the best found objective value (***BF***), convergence trend and running time. Because of the randomness in evolutionary algorithms, our experiments were performed 30 times independently with different random seeds on a computer Intel Core i7-8700 CPU 2.80GHz and 16GB RAM. All source code was written in Java programming language.

For an impartial comparison between algorithms, we set the maximum number of function evaluations to be $100000 \times K$, where K is the number of problems solved concurrently. The parameters of other algorithms such as SPT, LAST, and ESMFA were installed as in the original paper to achieve the best performance. Besides, we ran several tests to find the set of optimal parameters for our proposal. The parameters of P-MFEA are summarized in Table II.

TABLE II
PARAMETER SETTINGS FOR THE PROPOSED ALGORITHM P-MFEA

Parameter	Value
population size of each task (N)	200
number of potential individuals (NE)	75
distribution index of SBX crossover (μ_c)	2
index of polynomial mutation (μ_m)	5
Δt	150

C. Results and discussion



(a) RPD in MECAT problem

(b) RPD in MECAT-RN problem

Fig. 8. The relative percentage difference (RPD) between the average costs of compared algorithms

1) The averaged and the best objective value: Table III lists the averaged and the best found objective value (*Avg* and *BF*) obtained by algorithms on **Type 1** and **Type 2** datasets over 30 independent runs. The best averaged results are indicated in bold font. These results show that P-MFEA outperforms baseline algorithms on all instances in both datasets. Furthermore, the comparison in details of two problems is indicated by using the difference between the average costs of two algorithms (RPD) measurement [20]. The results of the MECAT and MECAT-RN problems are shown in Figure 8(a) and Figure 8(b), respectively.

In the MECAT problem, the largest RPD (P-MFEA, SPT) is 11.11% on instance m_120, and the instance with the smallest RPD is l_130 with a difference of 3.42%. It can be seen that P-MFEA outperforms the 2-approximation algorithm in both data types, especially **Type 1**. Besides, the maximum RPD(P-MFEA, ESMFA) is 2.58% on the m_120 instance and the lowest one is 0.56% on m_100. ESMFA has shown effective when providing good solutions close to an optimal value in [15], but the solution achieved by P-MFEA showed a significant improvement compared to ESMFA.

In the MECAT-RN problem, we investigated the relative difference between our proposed algorithm's archived results

TABLE III
RESULTS OBTAINED BY P-MFEA AND BASELINE ALGORITHMS IN TWO PROBLEMS: MECAT AND MECAT-RN

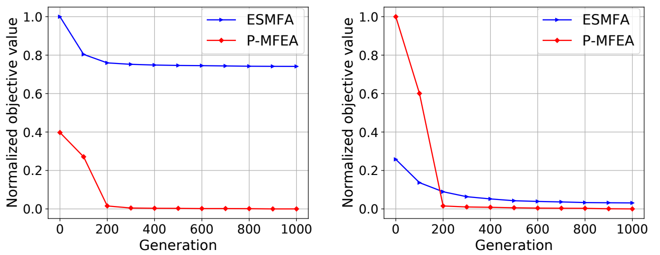
Instance		MECAT						MECAT-RN			
		SPT		ESMFA		P-MFEA		LAST		P-MFEA	
		Avg	*BF*	*Avg*	*BF*	*Avg*	*BF*	*Avg*	*BF*	*Avg*	*BF*
Type 1	*m_100*	453	453	418.7	414	**416.3**	411	351	351	**314.1**	309
	m_110	525	525	489.1	483	**479.3**	477	396	396	**341.4**	339
	m_120	540	540	492.7	486	**480**	477	408	408	**354.5**	351
	m_130	603	603	563.3	555	**551.9**	549	456	456	**400.5**	399
	m_140	690	690	636.90	630	**625.4**	621	537	537	**483.9**	480
	m_150	681	681	632.8	639	**622.5**	615	516	516	**464.4**	459
	m_160	765	765	710.9	717	**696**	693	591	591	**504.6**	501
	m_170	771	771	709.1	717	**695.9**	690	597	597	**524**	519
	m_180	828	828	776.7	783	**762**	762	612	612	**546**	546
	m_190	918	918	843.50	855	**827.1**	819	690	690	**620.6**	615
Type 2	*l_100*	675	675	654.5	645	**646.3**	642	558	558	**490.1**	486
	l_110	834	834	802.3	795	**793.5**	786	705	705	**587.2**	582
	l_120	846	846	819.5	810	**813.1**	807	723	723	**589.9**	585
	l_130	1017	1017	990.9	978	**982.2**	975	843	843	**713.7**	708
	l_140	969	969	924.1	909	**913.7**	909	888	888	**717.7**	711
	l_150	1104	1104	1055.9	1065	**1041**	1032	939	939	**786.1**	783
	l_160	1158	1158	1114.8	1137	**1103**	1095	945	945	**847.4**	840
	l_170	1173	1173	1132.3	1155	**1120**	1113	1071	1071	**813.7**	810
	l_180	1293	1293	1248.1	1266	**1236**	1230	1086	1086	**897.3**	891
	l_190	1311	1311	1259.4	1275	**1240**	1230	1191	1191	**936.8**	930

and one of the 7-approximation algorithm. In both datasets, RPD(P-MFEA, LAST) was greater than 10%. Particularly, the maximum and minimum values recorded are 24.02% (on instance l_170) and 10% (on instance m_150). These results demonstrated that the proposed algorithm was superior to the LAST algorithm. The reason is that the LAST has a relatively large approximation coefficient of 7, while our P-MFEA can find optimal or near-optimal results for the MECAT-RN problem.

In summary, the proposed P-MFEA proved excellent performance for both problems on both data types. For MECAT, the proposed algorithm showed the best improvement on the **Type 1** dataset, which has dense input graphs. Meanwhile, for MECAT-RN, P-MFEA exposed the most significant improvement on the **Type 2** dataset, which has a sparse input graphs.

2) Convergence trend: The average convergence trend of the two evolutionary algorithms P-MFEA and ESMFA was also compared to further evaluate the proposed algorithm's effectiveness. Due to space limitations, two instances representing each data type are shown in Figure 9. In these figures, the x-axis represents the number of evaluated generations and the y-axis represents the averaged objective value normalized by the formula in [16].

It can be seen in Figure 9(a) that P-MFEA exhibits excellent performance compared to ESMFA on the dataset of Type 1



(a) Instance m_100 (b) Instance l_190

Fig. 9. Convergence trends of two multifactorial evolutionary algorithms ESMFA and P-MFEA

in which the convergence trend of P-MFEA is much faster than ESMFA across all generations. Figure 9(b) depicts the convergence of the two algorithms on an instance of the Type 2 dataset. Although ESMFA's convergence is fast in the early stage of evolution, it could meet the local optima. In contrast, P-MFEA shows a fast convergence trend in the later stage when using the strategy of adding potential individuals to the existing population. It can be concluded that P-MFEA has an apparent efficiency when compared with the existing multifactorial evolutionary method.

3) Running time: Besides finding high-quality solutions, execution time is also an essential factor in solving opti-

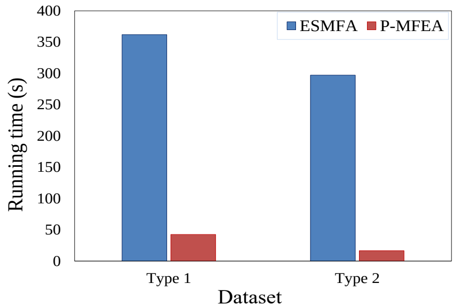


Fig. 10. Running time of EAs for MECAT problem

mization problems. To clarify the proposed algorithm's effectiveness, we compare the average execution time of three algorithms on the instances in two types of datasets. The running time of the two evolutionary algorithms, i.e., ESMFA and P-MFEA for solving the MECAT problem is depicted in Figure 10.

The fact that the running time of the 2-approximation algorithm is the fastest. However, its performance is ineffective compared to the other methods in terms of the solution's quality. When compared to ESMFA, our proposed algorithm provided faster running time. It can be explained that ESMFA uses genetic operators on tree structures, which causes a time-consuming transition from genotype to phenotype. In contrast, P-MFEA performs genetic operators directly on genotype. Moreover, adopting the strategy of adding potential individuals helps the search process quickly to find a good solution. In summary, P-MFEA demonstrates its effectiveness in terms of good solution quality and execution time when solving the problem of maximizing the wireless sensor network's lifetime.

V. Conclusion

In this paper, we study the problem of constructing the optimal data aggregation tree to maximize network lifetime for two types of networks: without relay nodes network and using relay nodes network. In addition, we propose a meta-heuristic, called P-MFEA, to address two problems simultaneously. The simulation results show that our proposed algorithm provides a good performance. As for future work, we plan to extend this work for three-dimensional networks such as underwater sensor networks and study the impact of node mobility. In addition, we will consider other encoding methods for our proposed algorithm. Furthermore, we also intend to extend the same for the heterogeneous wireless sensor networks in contrast to the proposed scheme which works only for homogeneous WSNs.

Acknowledgment

Nguyen Thi Tam, VINIF.2020.TS.105 was funded by Vingroup Joint Stock Company and supported by the Domestic Master/ PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), Vingroup Big Data Institute (VINBIGDATA).

References

[1] S. H. Lee, S. Lee, H. Song, and H. S. Lee, "Wireless sensor network design for tactical military applications: Remote large-scale environments," in *MILCOM 2009-2009 IEEE Military communications conference*. IEEE, 2009, pp. 1–7.

[2] A. Baggio, "Wireless sensor networks in precision agriculture," in *ACM Workshop on Real-World Wireless Sensor Networks (REALWSN 2005), Stockholm, Sweden*, vol. 20. Citeseer, 2005, pp. 1567–1576.

[3] A. Darwish and A. E. Hassanien, "Wearable and implantable wireless sensor network solutions for healthcare monitoring," *Sensors*, vol. 11, no. 6, pp. 5561–5595, 2011.

[4] D. Basu, G. Moretti, G. S. Gupta, and S. Marsland, "Wireless sensor network based smart home: Sensor selection, deployment and monitoring," in *2013 IEEE Sensors Applications Symposium Proceedings*. IEEE, 2013, pp. 49–54.

[5] Y. Sang, H. Shen, Y. Inoguchi, Y. Tan, and N. Xiong, "Secure data aggregation in wireless sensor networks: A survey," in *2006 Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06)*. IEEE, 2006, pp. 315–320.

[6] D. Mantri, N. R. Prasad, and R. Prasad, "Bhcda: Bandwidth efficient heterogeneity aware cluster based data aggregation for wireless sensor network," in *2013 international conference on advances in computing, communications and informatics (ICACCI)*. IEEE, 2013, pp. 1064–1069.

[7] S. Ramachandran, A. K. Gopi, G. V. Elumalai, and M. Chellapa, "Redd: Redundancy eliminated data dissemination in cluster based mobile sinks," in *2011 International Conference on Recent Trends in Information Technology (ICRTIT)*. IEEE, 2011, pp. 330–335.

[8] B. S. Mathapati, S. R. Patil, and V. Mytri, "Energy efficient reliable data aggregation technique for wireless sensor networks," in *2012 International conference on computing sciences*. IEEE, 2012, pp. 153–158.

[9] J. Yuea, W. Zhang, W. Xiao, D. Tang, and J. Tang, "Energy efficient and balanced cluster-based data aggregation algorithm for wireless sensor networks," *Procedia Engineering*, vol. 29, pp. 2009–2015, 2012.

[10] I. Mosavvar and A. Ghaffari, "Data aggregation in wireless sensor networks using firefly algorithm," *Wireless Personal Communications*, vol. 104, no. 1, pp. 307–324, 2019.

[11] A. Avokh and G. Mirjalily, "Dynamic balanced spanning tree (dbst) for data aggregation in wireless sensor networks," in *2010 5th international symposium on telecommunications*. IEEE, 2010, pp. 391–396.

[12] P. Patil and U. Kulkarni, "Svm based data redundancy elimination for data aggregation in wireless sensor networks," in *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2013, pp. 1309–1316.

[13] ——, "Delay efficient distributed data aggregation algorithm in wireless sensor networks," *International Journal of Computer Applications*, vol. 69, no. 1, 2013.

[14] T.-W. Kuo, K. C.-J. Lin, and M.-J. Tsai, "On the construction of data aggregation tree with minimum energy cost in wireless sensor networks: Np-completeness and approximation algorithms," *IEEE Transactions on Computers*, vol. 65, no. 10, pp. 3109–3121, 2015.

[15] N. T. Tam, T. Q. Tuan, H. T. T. Binh, and A. Swami, "Multifactorial evolutionary optimization for maximizing data aggregation tree lifetime in wireless sensor networks," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II*, vol. 11413. International Society for Optics and Photonics, 2020, p. 114130Z.

[16] A. Gupta, Y.-S. Ong, and L. Feng, "Multifactorial evolution: toward evolutionary multitasking," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 343–357, 2015.

[17] F. Rothlauf, "Representations for genetic and evolutionary algorithms," in *Representations for Genetic and Evolutionary Algorithms*. Springer, 2006, pp. 9–32.

[18] K. Deb, R. B. Agrawal *et al.*, "Simulated binary crossover for continuous search space," *Complex systems*, vol. 9, no. 2, pp. 115–148, 1995.

[19] X. Yu and M. Gen, *Introduction to evolutionary algorithms*. Springer Science & Business Media, 2010.

[20] P. Pop, O. Matei, and C. Pintea, "A two-level diploid genetic based algorithm for solving the family traveling salesman problem," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2018, pp. 340–346.